

Association for Information Systems AIS Electronic Library (AISeL)

ICIS 2009 Proceedings

International Conference on Information Systems
(ICIS)

2009

An Exploratory Case Study of the Benefits of Business Rules Management Systems

Chen Zhang

University of Memphis, czhang12@memphis.edu

Thomas O. Meservy

University of Memphis, tmeservy@memphis.edu

E. Ted Lee

University of Memphis, elee@memphis.edu

Jasbir Dhaliwal

University of Memphis, jdhaliwl@memphis.edu

Follow this and additional works at: <http://aisel.aisnet.org/icis2009>

Recommended Citation

Zhang, Chen; Meservy, Thomas O.; Lee, E. Ted; and Dhaliwal, Jasbir, "An Exploratory Case Study of the Benefits of Business Rules Management Systems" (2009). *ICIS 2009 Proceedings*. 19.

<http://aisel.aisnet.org/icis2009/19>

This material is brought to you by the International Conference on Information Systems (ICIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in ICIS 2009 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

AN EXPLORATORY CASE STUDY OF THE BENEFITS OF BUSINESS RULES MANAGEMENT SYSTEMS

Completed Research Paper

Chen Zhang

University of Memphis
czhang12@memphis.edu

Thomas O. Meservy

University of Memphis
tmeservy@memphis.edu

E. Ted Lee

University of Memphis
elee@memphis.edu

Jasbir Dhaliwal

University of Memphis
jdhaliwl@memphis.edu

Abstract

Business rules engines and business rule management systems (BRMS) are gaining popularity especially in large, complex, and real-time business environments. In essence, the business rules approach represents a new paradigm for compartmentalizing the formalization of business policies and rules as a separate component from application code. Given that this phenomenon has not yet been empirically investigated from a research perspective, we report on an exploratory case study undertaken to better understand the impacts of the business rules approach on activities throughout the software development lifecycle at a large Fortune 500 corporation. Our study integrates theoretical notions from the literature on knowledge reuse and systems theory with four constructs – centralization, standardization, externalization and structuration – to categorize the benefits arising from BRMS. Our results suggest that the application of the business rules approach has the potential to facilitate information systems development activities at all stages of the software development lifecycle.

Keywords: business rules, business rules management systems, software development, centralization, externalization, standardization, structuration, case study

Introduction

Business rules are central to most software deployed in organizations. These rules are “discrete operational business polic[ies] or practice[s]” (Ross 1994) or statements that influence and guide behavior in organizations (Steinke et al. 2003; Von Halle et al. 2006). Morgan (2002) defines a business rule as “a compact statement about an aspect of a business” (p. 5) and argues that the rule is expressed in simple, unambiguous language that may be understood by all interested parties. Essentially, business rules are “conditions that govern a business event so that it occurs in such a way that is acceptable to the business” (Von Halle 2002). Similarly, Ross (2003) defines business rules as “the encoded knowledge of your business practices” (p. 186). Typically these business rules are defined in requirements specifications and then translated into lines of code in software.

Currently, in many organizations business rules are still embedded in the myriad lines of code associated with various applications throughout the organization. The scattered and sometimes duplicated rules make it difficult for information systems development personnel to evaluate and maintain the integrity and consistency of rules. When errors or inconsistencies are detected, the distributed implementation of rules makes it more difficult to identify sources of errors. Furthermore, the various forms and systems in which rules are embedded and duplicated often require duplicated development, testing, and maintenance efforts.

When a business rule needs to be changed, it is a labor intensive, time consuming and costly process to pinpoint the exact location of the code containing the rule and to update it. Furthermore, business rules are often considered one of the most volatile parts of business software systems (Wan-Kadir et al. 2004). In today’s rapidly changing business landscape, the reality that organizations may need to make constant changes in business policies and rules further aggravates the inefficiencies caused by the traditional tight-coupling between business rules and code. In addition, globalized organizations often require different sets of localized business policies and rules in order to comply with the rules and regulations in foreign countries, leading to increased quantity and complexity of business rules that have to be codified.

In addition, the traditional tight-coupling between business rules and code also leads to problems in communication and collaboration among business owners, business analysts, developers, and testers. When rules are embedded in code, business owners who have provided the initial rule specifications tend to become less involved in the software development process because understanding the artifacts and materials used in the later phases often requires specialized training and technical expertise. On the other hand, in the design, development, testing, and maintenance phases analysts, developers, and testers rely on mainly descriptive documents, the interpretation of which is subjective and dependent on an individual’s background and expertise (Morgan 2002). Hence, due to the lack of business owners’ involvement and reliance on descriptive documents in the design, development, testing, and maintenance stages, misinterpretation and misunderstanding of business rules by analysts, developers, and testers may not be detected until later in the process, when mistakes are much more costly to rectify than earlier in the process.

An approach that holds much promise to overcoming these challenges is the business rules approach which decouples business rules from software applications that may be scattered throughout the organization. This approach externalizes business rules in a way that is unambiguous and understandable to business users, managers and technical personnel, resulting in a common business rules repository shared across applications and user interfaces (Von Halle 2002). A number of practitioner-oriented books about the business rules approach have been published (Morgan 2002, Ross 2003, von Halle 2002). Despite the increasing advocacy by practitioners and the growing popularity among organizations, the business rules approach remains under-investigated by information systems scholars. Although a few researchers have touched upon this topic from a technical perspective (Barrett et al. 2004; Jeng et al. 2004; Nagl et al. 2006), little is known about organizational adoption of the approach and the resulting impacts and even less is understood from a theoretical standpoint. In this case study, we examine the adoption of the business rules approach in an organizational setting and identify its impact on various stages of the software development lifecycle (SDLC).

The next section reviews relevant literature and sketches out the evolution of the business rules approach. The subsequent section introduces business rules management systems including four main theoretical constructs (i.e., centralization, externalization, standardization, and structuration) that may help to explain the perceived positive impact of BRMS on different stages of the software development lifecycle. A case study is then introduced and qualitative results are presented. These results highlight some of the perceived impacts of BRMS on the SDLC as reported by stakeholders with varying roles and titles that were associated with the project.

Literature Review

In this section, we first present some historical perspectives that are relevant to the business rules approach including how business rules are used in a traditional software development environment. We then review the literature on knowledge reuse and knowledge repositories.

Historical Perspectives

Systems architecture and general approaches for structuring software systems have evolved to handle the increased demand for more complex processing needs. According to systems theory, organizational business systems can be viewed as a complex, hierarchically nested system of production using concepts of modularity (Simon 1962). Each component of the system is, in turn, a subsystem of finer components. Modular systems theory posits that challenges in managing complex software systems can be partially addressed through the use of modular design principles such as loose coupling among components, information hiding (or encapsulation) within components, and compliance with standardized interface and performance specifications for modules (Baldwin et al. 1997; Baldwin et al. 2000).

In addition, the business rules approach is also related to rule-based expert systems, which usually consist of four components: a knowledge base, an inference engine, an agenda/scheduler, and working memory (Turban et al. 2006). A specialized subfield in artificial intelligence focuses on how to represent and encode knowledge into a machine-readable format. One of the most widely used knowledge representations in expert systems and knowledge based systems is called a production rule that links preconditions or antecedents to consequences. The production rules of these early systems were used to capture human expertise and were organized and managed in a knowledge base which was tailored to a specific domain. However, the first wave of commercial expert systems faded in the late 1980s and early 1990s due to a variety of factors such as lack of acceptance by users (Gill 1995), shifts in organizational priorities (Gill 1995), and lack of top management support (Duchessi et al. 1995). Yet, intriguingly, in the last few years there has been a resurgence of interest in using rule-based systems. Many software vendors, such as IBM and Oracle, have recently invested heavily in developing sophisticated business rules management products. In this research, we propose to re-examine business rules in software development from a knowledge management perspective because business rules are, in essence, a form of basic business knowledge. Business rules control or influence the behavior of the business by describing the operations conducted by the business and the constraints adopted by the business. They represent “the terms, facts, and rules of the business” (Ross 2003, p. 90) and are often represented in the form of organizational decisions and policies. Business rules are central to most software deployed in a business or an organizational environment because they “govern a business event so that it occurs in such a way that is acceptable to the business” (Von Halle 2002).

In a traditional software development environment, business rules are initially defined and specified separately. These rules are often encoded in plain or structured natural language as part of a requirements specification that is created during the analysis and design phases of the SDLC. During the implementation phase, these rules are then codified in the form of programming-language specific syntax within the applications that are distributed throughout the organization. Rules that are specified in requirements are often duplicated within the different programs throughout the organization. Selective changes in particular contexts and applications can also render them inconsistent over time. From a knowledge management perspective, rules hard-wired in applications scattered throughout the organization lead to limited access to and understanding of the rules and more importantly, a lack of rule reuse. There is relatively little research targeted at understanding knowledge reuse as it pertains to business rules.

Knowledge Reuse and Knowledge Repository

Given that knowledge is one of the most important sources of competitive advantages for organizations, many organizations have spent an enormous amount of effort trying to find ways to better manage their knowledge resources. One main objective of knowledge management is to achieve effective reuse of existing knowledge.

The knowledge reuse process consists of capturing, packaging, disseminating, and reusing knowledge (Alavi 1999; Markus 2001). Knowledge may be captured “through written form, spoken presentations, or discussions given by or to individuals or groups” (Bednar 1999). Knowledge can be captured and documented as a by-product of the work process, or “within a structure such as that provided by facilitators using brainstorming techniques”, etc. (Markus

2001). According to Myers and Swanborg (1999), knowledge packaging translates and structures information into usable knowledge. Some packaging activities include codifying knowledge into knowledge objects by providing context, filtering and pruning content, and authoring knowledge content (Galunic et al. 1999; Roth et al. 1998). Knowledge is then disseminated and made available to others. As pointed out in Markus (2001), knowledge disseminating includes activities such as publishing a newsletter, adding content to a repository, or helping others use knowledge effectively. The last stage of the knowledge reuse process is reuse, which involves both recall (where information is stored) and recognition (information satisfies the user's need and the user applied the knowledge). Markus (2001) further identifies three roles in the knowledge reuse process, namely knowledge generator or producer, knowledge intermediary, and knowledge consumer. Based on the type of knowledge reuser and the purpose of reuse, Markus (2001) categorizes knowledge reuse into four types: reuse by shared knowledge producers, reuse by shared work practitioners, reuse by expertise-seeking novices, and reuse by secondary knowledge miners. She also argues that the more dissimilarity between knowledge reusers and generators, the more difficult it is for reusers to identify the question, locate expertise, and reuse knowledge.

Information technology can provide tools that act as an intermediary by “automatically categorizing, abstracting, filtering, and disseminating documents” (Markus 2001). Among these IT-enabled tools are knowledge repositories (Markus 2001), or organizational memory systems (Ackerman 1996), or organization memory information systems (Hackbarth et al. 1999). Repositories can store external knowledge, structured internal knowledge, and informal knowledge (Davenport et al. 1998) and may play a facilitating role in almost all knowledge reuse situations (Markus 2001).

Business Rules Management Systems

Business Rules Management Systems (BRMS) are software environments where business rules for an organization are defined, deployed, executed, and monitored. Core to business rules management systems is the concept that the decision logic of an organization (e.g., policies, requirements, and conditional statements) can be codified and stored in a centralized location rather than being dispersed and duplicated in disparate applications scattered throughout the organization. In general, BRMS facilitates deployment of applications in a service-oriented architecture (SOA), where the rule engine, a key component of BRMS, presents itself as a service to applications and applications themselves represent deployable services such as web services. Figure 1 displays a simplified representation of the status quo (i.e., non-BRMS) environment and also a BRMS enabled environment. Four main characteristics differentiate the business rules approach from the status quo: externalization, centralization, standardization, and structuration.

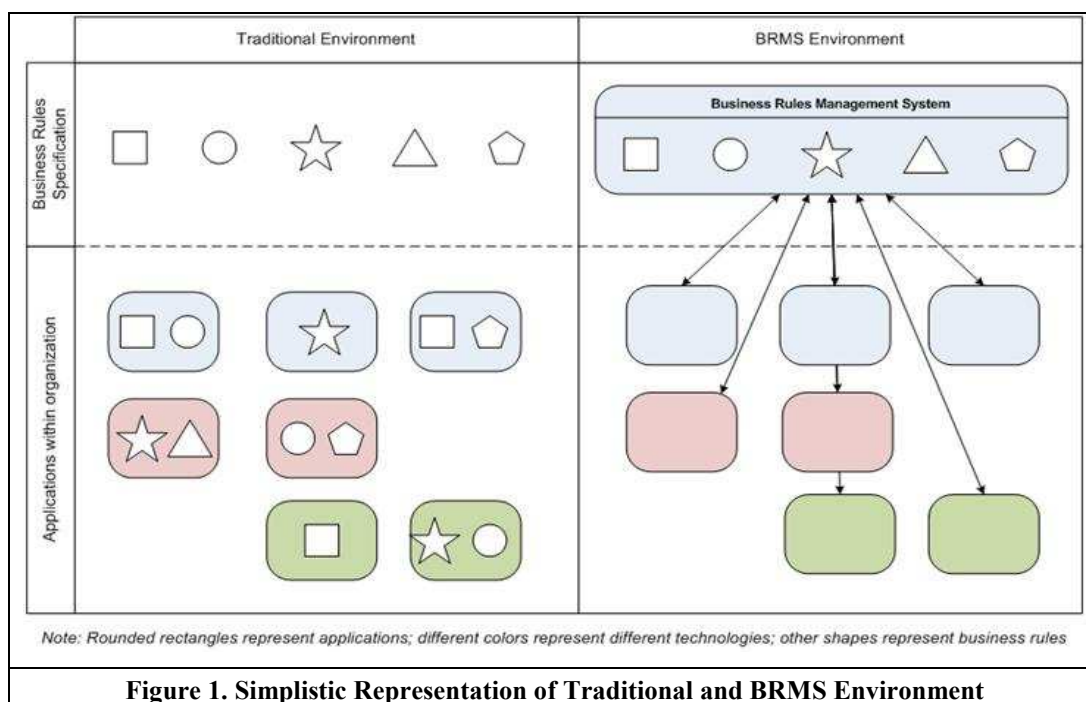


Figure 1. Simplistic Representation of Traditional and BRMS Environment

Externalization refers to the separation of rules from processes and applications. It is the key to achieving the benefits of the business rules approach. Our definition of externalization differs from that of knowledge externalization, which refers to the conversion from tacit knowledge to explicit knowledge (Nonaka 1994). Hendriks (1999) points out that knowledge externalization may not be a conscious act intended to facilitate knowledge sharing. In traditional software development business rules are specified in requirement documents and then converted to software code. However, such conversion does not necessarily aim at sharing business rules with others. Due to the technical barriers caused by the growing complexity of software and difficulty of understanding software code, others are often unable to find and understand the codified business knowledge embedded in the code. Therefore, by separating business rules from software code, business rules management systems facilitates knowledge sharing among individuals with various expertise and background. De-coupling rules from code also results in smaller and simpler code, leading to less development effort and shorter development cycles, and allows for enhanced agility and flexibility for BRMS-enabled organizations. Externalization also makes it possible to achieve centralization of business rules.

Centralization refers to the storage and management of business rules in a centralized location – business rule repository. Having a centralized repository to store corporate data has a long standing history in database area. Benefits of the centralized approach for data include: data independence, improved data consistency, improved data sharing, improved productivity of application development, enforcement of standards, improved data accessibility and responsiveness, etc. (Hoffer et al. 2009). An extension to this idea is the concept of data warehousing where data is taken from multiple sources and stored in a centralized repository to facilitate decision making (Cooper et al. 2000; Inmon 2005). Business rule repositories avoid duplication of rules in disparate applications. Due to centralization of rules, BRMS reduces the number of times a rule is implemented across the organization (e.g., separate applications, different technologies) and subsequently reduces the maintenance costs of these applications when updates to business rules are required.

Standardization refers to the way in which the rules are defined in a common, consistent representation that multiple stakeholders understand. Standardizing the representation of knowledge can be traced to the artificial intelligence research that explores the use of formal ontologies to facilitate knowledge sharing and reuse among software entities (Gruber 1995). According to the business rules approach, rules are expressed in "clear, unambiguous, well-structured business English" (Ross 2003) in a standardized format. The standardization of rule expressions in business English allows non-IT personnel (e.g., marketing, business analysts) to understand, define and modify business rules whereas rules embedded in code are not easily understood by non-IT personnel. The unambiguous expression of rules also improves cross-functional communications within the organization and across the SDLC.

Structuration refers to the connection of a rule with other rules. In a more general sense, structuration also includes an understanding of where a rule originates from and where it is utilized. Similar to database schemas that define the relationships of data and structure of databases, the relationships among business rules are explicitly specified. Rules are represented in a way that maintains a connection for each rule in two directions, one direction toward its origins (e.g., business goals, strategies, and tactics) and the other direction towards its implementation (Von Halle 2002). Rule structuration not only provides the ability to easily monitor rule execution to quantify the frequency, volume, and other aspects of rule execution but also enables impact analysis of rule changes through rule chaining.

These four characteristics that differentiate BRMS from traditional software development environment can also be mapped back to the knowledge reuse process introduced earlier as displayed in Figure 1. Knowledge sources external to the company, as well as internal sources are used by producers to generate new knowledge in the form of business policies and rules. These rules are then captured, packaged and disseminated utilizing a technological intermediary. In traditional software development business rules are captured in the form of requirements documents, packaged in the form of software code, and disseminated in applications, making it difficult for non-technical personnel to recall (where knowledge is stored) and recognize (knowledge satisfies the need and is reused) in the knowledge reuse process. In contrast, in BRMS-enabled environment externalization, or decoupling of the rules from the code, provides the opportunity to centralize once disparate rule sets. Additionally, the visibility of standardized and structured rules is made possible by the externalization and centralization of the rules. Business rules are captured and packaged in a standard, English-like language, stored in a centralized repository and made available to all the people involved in the software development, facilitating reuse of core elements of business knowledge represented as business rules.

According to Gartner, BRMS emerged as an integration of Business Rules Engines (BRE) and Business Process Management (BPM) products. Many BPM products contain a rule execution engine, a rule repository, rule

modeling and simulation, rule usage monitoring and analysis, rule management and administration, rule templates, and an integrated rule development environment (Gartner 2008). Existing BRMS products provide these functionalities to various extents. Some open source BREs such as JBOSS Drools and NxBRE mainly provide the rule engine functionality. Many commercial vendors such as Fair Isaac and ILOG have provided or are migrating to the full range of functionality. Among the commercial products, Pegasystem's rules engine technology provides a unification of business rules with business processes. Though there are some differences in the features that each of these BRMS contain, there is sufficient similarity that organizations utilizing any of these BRMS implementations would obtain the benefits of centralization, externalization, standardization, and structuration assuming the technology was correctly appropriated.

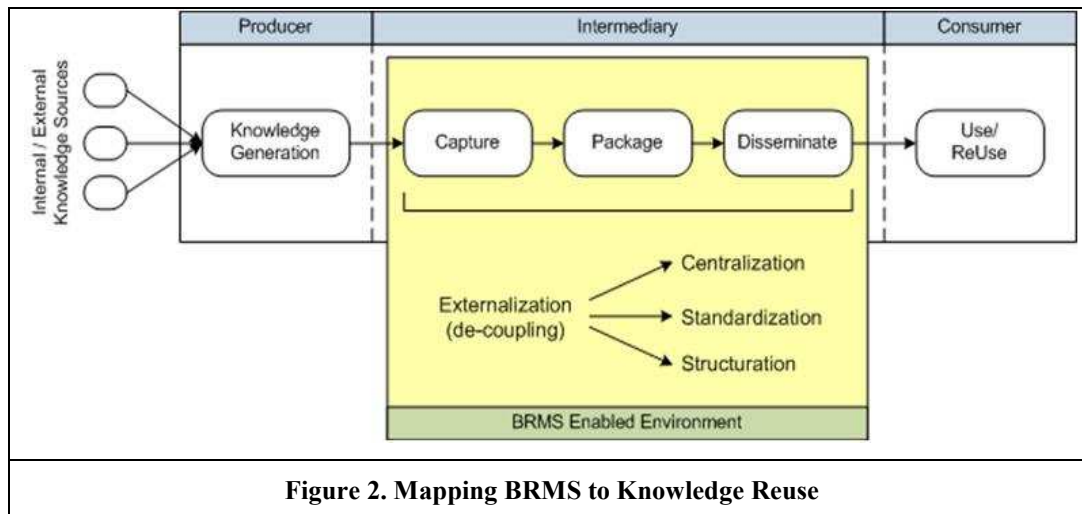


Figure 2. Mapping BRMS to Knowledge Reuse

However, despite organizations' and practitioners' rising interest in the business rules approach and BRMS products (Anonymous 2007; Rymer et al. 2008), few researchers have investigated the impact of the business rules approach at different stages of the software development life cycle. To further explore the impacts of BRMS and the business rules approach, we conducted a case study at a large international logistics company where a subdivision recently adopted and implemented a BRMS.

Research Method

Case Context

FedEx is a Fortune 500 company that has a very diverse and complex software application portfolio. This portfolio of applications consists of customer facing applications, middleware applications and backend systems that need to operate in concert in order to provide the best customer experience. This entire fleet of applications is mission-critical and any 'down time' directly impacts corporate revenues and customer satisfaction.

Since these applications have been developed and maintained by multiple individuals, the core business knowledge about the system is spread across various documents, applications, and individuals. Writing requirements for a new feature often requires extensive research into the current behavior of applications and business rules. The lack of a central, unified, and accurate business knowledge base that requirements writers can rely on leads to potentially ambiguous or incomplete requirements. This problem, in turn, can translate into inaccurate implementation, ineffective testing, and ultimately a product that may not accurately meet the original need of the customer.

Adoption and Implementation of BRMS in the Shipping Rules Team

One way to break away from the vicious cycle of complex and ambiguous requirements to ineffective testing is to centralize the knowledge about business policies and business rules into a single location. This will make core business knowledge accessible to all stakeholders whenever they need it. After preliminary research by a small team of employees, BRMS was identified as a potentially viable solution. During the rigorous evaluation stage, this team was tasked to start detailed research on the technology, including consultation with Gartner and Forrester to

learn the trends, leading vendors, adoption risks, etc. As a next step, the team started experimenting with some of the leading BRMS products available on the market by conducting a proof of concept to ensure that the product had the potential to solve the problem at hand.

Then a more formal and detailed evaluation phase of the technology was initiated. The goal was to define the key metrics for selecting a BRMS product out of 3 to 4 commercially available products. The selected criteria consisted of scalability, performance, feature support, costs, and risks. At the end of this evaluation phase, one BRMS product was selected.

The Shipping Rules Team pioneered the use of BRMS at FedEx by implementing shipping business logic using a phased approach. During the first phase, Domestic Shipping rules were coded and applications were rewritten, which took a number of months. During the second phase, a more challenging set of International Shipping Rules were coded and implemented in the BRMS product. Each phase followed a disciplined process as recommended by the business rules approach. Comprehensive testing and incremental development models ensured that this migration to new application was as transparent as possible to their customers.

The broad timeline of the achievements was: tool evaluations (six months), technology pilot and evaluation (twelve months), Phase One: Domestic Shipping Implementation (nine months), and Phase Two: International Shipping Implementation (nine months). Finally, a production deployment was launched and functional migration is currently underway. The Shipping Rules Team in FedEx has moved the majority of its business rules from conventional code to BRMS technology and centralized the core shipping knowledge into the business rules repository.

Data Collection and Analysis

The authors participated in face-to-face meetings and audio-only conference calls with key members of the project team over a period of about 3 months. There were 10 informants including the manager over the functionally organized teams, four requirements writers, three members of development, and two testers. This set of individuals represents the main stakeholders involved in the BRMS project.

During the data collection period, the research team made multiple visits to the project team's offices and had multiple conference calls with them. Benefits were initially identified based on interview-style group discussions at the organization. Subsequently, more detailed interviews were conducted to further understand the perceived benefits from perspective of different roles. These multiple person, interview sessions typically lasted over an hour. Additionally, there were a few work sessions with the project team based on initial data analysis in order to solicit additional ideas and to confirm completeness and accuracy of analysis.

During this exploratory case study, data collection and data analysis overlapped. In line with grounded theory and following the pattern of previous case studies (Mao et al. 2008), our content analysis proceeded through three main steps. First, we identified major characteristics exemplified by BRMS and stages of the software development lifecycle. Then, in the second step, we classified salient benefits based on the impact of BRMS at different stages of the software development lifecycle. At this stage we also classified each of the benefits according to the main characteristic of BRMS that enables the specific benefit. Lastly, these detailed benefits were aggregated into more general groupings.

Results

The software development process involves the sequential generation of requirements, design, code, and implementable configurations. Based on our case study, this section describes the overall benefits of implementing a business rules approach and also the benefits to each phase of the development process. These benefits flow from the four main characteristics of the business rules approach as identified in previous section, namely centralization, externalization, standardization and structuration.

Impacts of BRMS on the Overall Software Development Lifecycle

There are many benefits of BRMS that are realized across multiple stages of the software development lifecycle. Perhaps the most frequently cited overall benefit was the idea that BRMS enabled easy access to rules. That is, there was a single repository where all of the business rules were stored and those rules were implemented in a way that both technical and non-technical personnel could understand. A single source of business rules helps

individuals navigate and find accurate information easily. Having core shipping rules in one central place has allowed all the business partners including requirements writers, developers, testers and other business users to access these rules. Not only does this impact requirements writing, which is further discussed in the next section, but it allows review of business rules and testing earlier in the development lifecycle. The accessibility of rules results in a drastic improvement from the previously implemented business logic that was buried deep in code understandable to developers only. This arrangement not only ensures that everyone is aware of his or her own part of the application but also helps the individual better understand how different parts fit into the overall application.

Another, frequently cited, overall benefit of BRMS is the emergence of a shared vocabulary. The definition and implementation of rules in a consistent, standardized format and storage of the rules in a location accessible to everyone facilitates cross-functional communication. Common terms and definitions enable teams to catch mistakes early in the development cycle

Information from interviews also revealed that centralized rules were easier to maintain and that overall maintenance costs were lower. Abstract rules that were once coded in multiple locations were now implemented in a single location, but the rule sets were reused for different configurations (e.g., both an online, web-based application as well as a stand-alone application that does not require the internet). This ability to consistently reuse the same implementation of the rules eliminates what the team termed as channel conflict that occurs when an end-user has a different experience depending on what channel of the product they utilize (e.g., web-based vs. stand alone). By implementing a BRMS, consistent results are delivered through various channels and applications that utilize the centralized set of rules.

Interestingly, the project team discovered that business rules stored in a centralized location and expressed in an English-like language provided assistance in employee training. The value of overall positive effects of implementing BRMS to achieve centralization, externalization, standardization, and structuration cannot be under-emphasized. Table 1 summarizes some of the benefits of BRMS on the overall software development lifecycle and also includes representative quotations from the case study.

Table 1. Summary of Overall Benefits					
Benefits	Representative Quotations	Centralization	Externalization	Standardization	Structuration
Consistent results are delivered through various channels and applications using a centralized set of rules	"We're offering access to business rules from various channels. So from the marketing perspective, we want our customers to get the same experience no matter which channel they are coming in. Whether that is through the Internet, web services, or client software that we have given them, whatever the case may be we want them to get the same experience and same answer. By externalizing the rules, we are able to give them that same experience and remain consistent."	X	X		
Centralized rules are easier to maintain at a lower cost	"So, say I have four channels and each of them had the rules hard-coded or coded within logic, well, I've got to have four teams maintain that code. Whereas if I externalize the rules, now I can have one team coding the rules and all four channels get the benefit of it. That is one team, one set of hardware, one set of everything. So we save a lot."	X	X		
Utilization of common vocabulary facilitates cross-functional communication to agree on terms and definitions, enabling teams	"I think that is huge. That is all it is about. About collaboration across all areas. Again, it forces conversations up front in the software development cycle to agree on terms and definitions that would, in a traditional environment, be [found] in the end. If you			X	

Benefits	Representative Quotations	Centralization	Externalization	Standardization	Structuration
to catch mistakes early in the development cycle	find a mistake [like] that [it] would be costly.”				
A single source of business rules helps individuals navigate and find the accurate information they need easily.	“What BRMS helps is that collaboration is not labor intensive. If I am researching the rules, I don’t have to call 5 people and go through 7 different documents. I can go to one place and browse, and that is all. So the labor-intensive collaboration has now become a very automatic collaboration, which is fairly accurate.”	X	X	X	X
Much of review and testing can be performed earlier in the development cycle.	“The interesting part of this is we know so many things up front. All the design work is done up front (such as consistent pattern). The code is very straight forward. We have done all the work up front. Typically you have the coding cycle. Typically, a week or two before turning the code to testing, that is when developers would have their walkthroughs. With BRMS, you probably still have that overview/ rule reviews. But you have additional reviews way up front, where you would not have in traditional [code].”		X		
Same rules base can be reused for different configurations.	“BRMS facilitates reuse of rules (international or domestic service).” “[Interviewer]: As you mentioned earlier, they see something and see patterns and then just duplicate that to the new scenario. How often are people finding this? [Developer]: A lot of situations. There are only a [limited] number of patterns in world shipping rules. When we get a new feature, it is sort of enhancement of existing one. When we get new rules, they tend to be similar because it is all about shipping. We can easily take one cookie cutter and apply it.”	X	X	X	
Business rules stored in a centralized location and expressed in English-like language provide assistance to employee training.	“One thing I’ve seen is when we transition people to that system, it’s a way to get up to speed quicker and understand the inner working of some of the business rules. It kind of goes back to the single source but it contributes to transition time. And I think that is across requirements, development, testing, etc., even marketing if we allowed them to view it.”	X	X	X	

Impacts of BRMS during Requirements Phase

Business systems requirements represent the first artifact that is developed in the early stages of the software development lifecycle. Often, these requirements first take shape as use cases and user narratives that explicate the features that the application being considered needs to contain. The language that business rules are expressed in is structured but easy to understand. The way rules are defined and structured allow requirements writers to quickly understand how rules are connected to other rules. These aspects help requirements writers understand the rules more quickly. Business analysts trained in business rules can participate in validation of requirements early on.

Business rules represent a good format for capturing business policies and logic because the providers of such requirements are generally comfortable with the IF-THEN-ELSE format. Additionally its logical, natural language like syntax is relatively easy for even non-technical users to quickly learn. The business rules approach facilitates capturing more complete and accurate requirements and also facilitates early and efficient testing of those requirements. Visibility into core shipping logic makes them understand how the various parts of the system are connected. Improvement in the quality of requirements has resulted in correct implementation and effective testing. The net impact is enhanced application quality and improved customer experience. Changes to business rules are traceable, enabling requirement analysts to better understand the history of rules. In addition, availability of complete and accurate source of business knowledge has assisted requirements writers in their research. A single source of business rules helps requirement analysts find the accurate information they need faster. Interviews revealed that this was one of the most salient benefits of BRMS to requirements writers. Table 2 below captures in detail various ways in which the requirements writing is facilitated by the business rules approach.

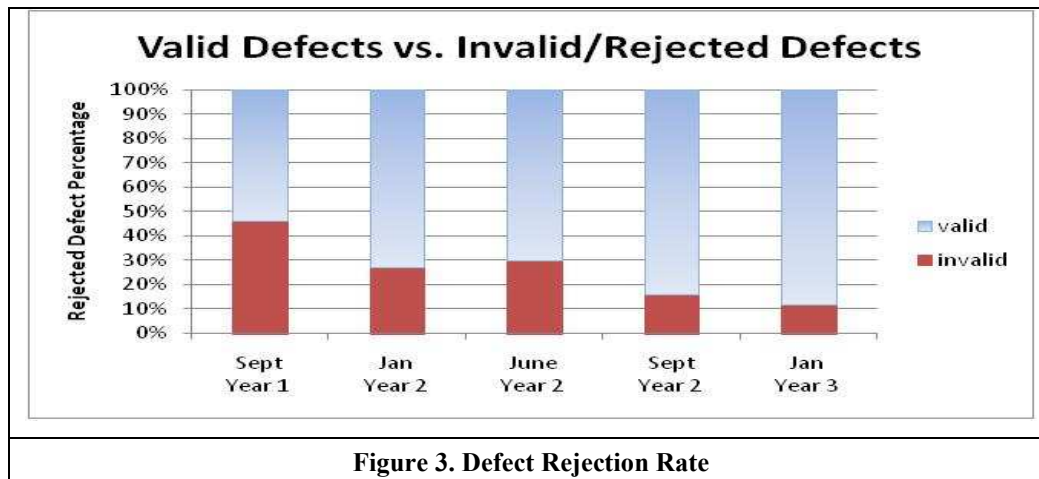
Table 2. Summary of Benefits for Requirements Phase					
Benefits	Representative Quotations	Centralization	Externalization	Standardization	Structuration
A single source of business rules helps requirement analysts find the accurate information they need faster.	<p>“It’s a single source of information. You don’t need to go to multiple places to get information. What we used to do was, we had two other sources of data that we could go get this information from. We would search by project and by product. ... I don’t have to go by project to get information. It’s just a one stop shop and it is of the quality that I am looking for. So it’s just one place that I need to go to. It’s accurate too.”</p> <p>“Whenever we get a new feature they (requirement writers) have to know how it impacts the existing one and they have to do a lot of research to find that out ... since they have all the existing business knowledge in one place their research time has reduced by about 20 to 30% from it was previously. So they could do research much faster, more accurate.”</p>	X	X	X	
Changes to business rules are traceable, enabling requirement analysts to better understand the history of rules.	“One of the issues that I think we had when we were trying to use documentation is that maybe something was implemented and maybe it wasn’t. It was changed in the future and how do you track those changes.”	X	X		
The language that business rules are expressed in is more readable, helping requirement writers understand the rules.	“Another thing that I wanted to mention was the language of the BRMS itself. It is more readable, it is like English. That helps us a lot. So the language itself of the BRMS is an advantage for us. You don’t need special training to understand what it means so it is easy to read and you can understand immediately.”			X	
Business analysts trained in business rules can participate in validation of requirements early on.	“Analysts can validate requirements based on business rules and catch [any] discrep[an]cies with rules before turning requirements to developers.”	X	X	X	

Impacts of BRMS during Design, Coding, Testing and Maintenance Phases

The design phase involves designing code and other artifacts that measure up to good standards, performance criteria, specifications from prior stages, and other requirements. The business rules approach provides positive impacts at this phase. The centralization of all the rules in a single rules repository and the standardized format of the rules facilitate reuse of business policy components across multiple applications and environments, making design validation faster and simpler. The rules based approach helps designers solve new problems using existing patterns and detect mistakes earlier. Our case study interviews also suggest that the standardization of the business rule approach assists in the early emergence of a consistent vocabulary in the form of an organization-wide domain object model (DOM) that facilitates program and data design.

During development, or the coding phase, the design artifact is implemented in programming language/platform-specific code. Many common coding errors often results from formalizing business rules and policies in code. Due to the externalization of the business rules from the program code, the design and the subsequent implementation of the code is simpler. This decoupling simplifies development. Modular units of code become smaller which makes both code reviews and unit testing less complicated and faster. An added benefit of this is that defect tracking and fixing is also simplified. Business rule reviews can be handled separately from code reviews and can involve participation of non-technical business users.

The testing of requirements, design, and code artifacts is facilitated in a BRMS environment. Due to the standardized nature of the business rules in the BRMS, information is interpreted in a less ambiguous way. In addition, due to the accessibility of the rules repository testers can access and understand business rules, which enable them to conduct research on rule-related defects independently without the need to consult with developers. Furthermore, some BRMS products provide the ability to visualize rules and relationships to other rules. Overall, access to the standardized, structured rules repository and the visual representation of the rules has enabled testers to truly understand the behavior of rules and helped them write more effective test scenarios, leading to better test coverage and higher test penetration. As a result, critical problems are more likely to be detected early in the testing cycle. In our case study, after integrating the BRMS, the defect rejection rate (an important metric that captures the effectiveness of testing) was reduced from 46% to 12% within the span of 16 months as illustrated in Figure 3.



The systems maintenance phase is often cited as one of the most costly phases of developing and implementing a system. Some researchers and practitioners suggest that as much as 60%-80% of the cost of an information system comes during this phrase (Valacich et al. 2009). Over time, business rules and other aspects of the system change which necessitates changes to the implemented systems. During this phase of the SDLC the BRMS approach continues to yield significant benefits. First, research time is decreased significantly. That is, the labor intensive, costly process of pinpointing rules implementations and how they are used in conjunction with other rules is simplified because the rules are visible and easily accessible to both technical and non-technical personnel. Secondly, the actual cost altering of the business rules yields tremendous savings because the implementation occurs in a single location rather than in multiple applications that may be distributed throughout the organization.

A summary list of benefits and representative quotations from our case study on the impact of BRMS on the design, coding, testing, and maintenance phases can be found in Table 3.

Table 3. Summary of Benefits for Design, Coding, Testing and Maintenance Phases					
Benefits	Representative Quotations	Centralization	Externalization	Standardization	Structuration
Design					
Business rules approach helps early definition of vocabulary (DOM), which facilitates data design	“We have the DOM and we view it up front and we give the feedback (if something is not good, you need to change it). It is part of the discipline which we maintain. BRMS does help that.”			X	
Rules based approach helps designers solve new problems using an existing set of patterns and detect mistakes earlier.	“What we do is we have standard cookie cutters or patterns to solve the problem. When we do the design, we say ‘oh this is the problem and we can use this pattern to solve it’. Once we apply that pattern, we know pretty much what the design will look like. If we see any problems, we can detect them right at that stage as oppose to finishing coding and two or three weeks down the line and catch it there. So there is the benefit of catching problems earlier.” “When developers do the coding, they can code it in so many different ways. You have zillions of ways to code it. Now it is more consistent. If you implement this one, use this pattern. And then the code has the similar pattern.”	X			
Coding					
Decoupling rules from software code results in less code and simplifies development.	“None of the business logic is in [programming-language specific code]. From that aspect, [the code] is simpler.” “We only have one door to enter BRMS instead of five or six doors and a few windows. You only have one door, one method call to BRMS.”		X		
Testing					
Testers can access and understand business rules, which enabled them to conduct research on rule-related defects independently without the need to consult with developers. As a result, a greater percentage of opened defects are valid.	“Defect management is facilitated because testers can search our business rules. That is a huge benefit to testing.” “There are some times when we find some defects and what do we do, before we send back those defects to development, we check if those requirements are perfect or not, based on BRMS, and we see whether this is really a defect itself and we can assign these defects back to them. You know, development does not also work on those invalid defects.”	X	X	X	
Standard, structured and visual representation of business rules facilitates test case design.	“BRMS is really helpful for us to predict our test cases and also if there is something missing we can add these cases in our test bank also.”		X		X

Benefits	Representative Quotations	Centralization	Externalization	Standardization	Structuration
Maintenance					
Centralization of business rules result in cost savings because fewer developers are needed to maintain software when business rules are updated.	“So, say I have four channels and each of them had the rules hard-coded or coded within logic, well, I’ve got to have four teams maintain that code. Whereas if I externalize the rules, now I can have one team coding the rules and all four channels get the benefit of it. That is one team, one set of hardware, one set of everything. So we save a lot.”	X	X		

Implications / Discussion

As discussed in the previous section, the majority of benefits that arise from utilizing BRMS technology and applications flow from capturing, packaging, and disseminating business knowledge in the form of business rules. These processes in turn lead to more effective knowledge reuse. Our study has attempted to empirically demonstrate the pragmatic value of the business rules approach to an actual business context. It has also shown that there is value to understanding the sources of benefits from a theoretical perspective using modeling/categorization constructs such as centralization, standardization, externalization and structuration that can serve as the basis for theory building. In our view, the decoupling that is promoted by the use of BRMS is in many ways not different from the modularity in design that was applied to data and process modeling during the era of structured systems analysis and design. The same benefits that arose from tightly cohesive and loosely coupled data or program modules are still relevant in relation to business rules. Their value is actually accentuated significantly because the business rule engines are linked to both development and operating environments on a real-time basis. The fact that enterprise-wide business knowledge is centralized in a tightly cohesive and loose coupled manner using easily accessible rules as its representational formalism certainly requires us to think differently about both knowledge management and software development.

Our study has important implications for research. It is one of the first studies that investigate organizational knowledge expressed in the form of business rules from a knowledge reuse perspective. It considers business rules as one important type of organizational knowledge and clarifies how such knowledge can be reused in multiple applications, leading to increased efficiencies and effectiveness in software development. We also identified and presented four major conceptual constructs underlying BRMS: centralization, externalization, standardization, and structuration. We incorporated previous knowledge management models into an extended model in order to tie our four conceptual constructs of BRMS to this extended model.

The findings of our study can be also beneficial to practitioners who are interested in adopting and implementing BRMS. The case study conducted at a large company with a very diverse and complex software application portfolio provides a ground-up view of the impacts of BRMS on various phases of systems development. It provides a detailed list of the benefits of BRMS at each of these phases. Among others, some salient ones include decreased research time, shared language and vocabulary, enhanced test coverage based on visibility and structuring of rules. We also suggest that the BRE approach is most useful for organizations that have a complex application portfolio where business rules are currently duplicated in multiple applications. The results of our study can assist those seeking a nuanced and careful assessment of the benefits of BRMS as part of a feasibility analysis.

Considering all the benefits of BRMS identified in our study, one might jump to the conclusion that BRMS are a silver bullet to the problems in capturing and maintaining complex and fast changing business knowledge. Even though there are numerous benefits to using a BRMS approach, there are also potential drawbacks.

Adopting the BRE approach may require changes to business processes in order to achieve process optimization because this approach is regarded as a paradigm shift in software development. Evelson et al (2008) even argue that

the BRE approach needs to be incorporated into new application architectures that include business process management and business intelligence for business optimization. New technologies and approaches such as BRMS often initially disrupt the status quo: personnel must work across functional areas, budgets may need to be reallocated (e.g., who is responsible for paying for the shared system), and from a political perspective a shift in power from one group to another may occur. Additionally, the business rules approach requires a different skill set from the development team and calls for a drastic shift in the mindset of those creating and maintaining software. For example, one team lead noted, "It is a fundamental paradigm shift. It's a totally different way of thinking about software because we used to really think in terms of [specific] languages, functions, or modules. Now, you really have to think in terms of rules, [which] is a fundamental shift." As with the adoption and implementation of other information systems, proper planning, education, application and training for users are important for successful adoption and implementation of BRMS.

Furthermore, business users of BRMS applications also need to change the way they think about accessibility to their application systems and business rules. No longer are these only accessible through technical intermediaries. These systems are now directly accessible for validation, analysis and training purposes as they can be viewed directly in the understandable format of business rules rather than technical programming code, functions and modules. It will be interesting to see how traditional and newer coding languages and platforms transition and change to take advantage of the new functionality and benefits offered by BRMS. This shift will require users to have a different set of skills and a different perspective, which will not only mandate careful planning for adoption and implementation but also newer and simpler methods for systems deployment.

It is also important to recognize that one limitation of our study is associated with the case study methodology that involves only a single organization and therefore may not be representative of the population, limiting the generalizability of our findings. However, given that business rules and reusing business rules have been under-investigated in IS research, the case study methodology can help researchers generate new thinking and provide great opportunities for conceptual and theoretical development in this domain. Future research should further evaluate and validate the impact of centralization, externalization, standardization, and structuration by examining these characteristics in other organizations and contexts.

Conclusion

Business rules are often considered one of the most volatile parts of software systems. Traditionally, business rules are embedded and duplicated in applications that are distributed throughout the organization. Businesses may experience inefficiencies in the implementing and testing rules that appear in these disparate applications. This study investigates the business rules approach, which decouples business rules from the software systems that are scattered throughout the organization. Business rules management systems standardize, externalize, centralize, and structure business rules in such a way that the rules are unambiguous and understandable to not only to technical personnel by also requirements writers, marketing and other less-technical personnel and also make them centrally accessible to. We conducted a case study at a subdivision of a large Fortune 500 corporation that recently implemented a BRMS product. As a result of the case study, we identified a number of benefits of using the business rules approach. These benefits highlight the impact of the approach on activities at each phase of the software development lifecycle. Results suggest that the application of the business rules approach has the potential to facilitate information systems development activities at all stages of the software development lifecycle.

Acknowledgements

This research was supported by the Systems Testing Excellence Program of the FedEx Institute of Technology at the University of Memphis. We would also like to acknowledge the contribution of Joel Tracy and the Automation Shipping Team at FedEx, as well as personnel in other departments who contributed to the case study.

References

- Ackerman, M. "Definitional and Contextual Issues in Organizational and Group Memories," *Information Technology & People* (9:1) 1996, pp. 10-24.
- Alavi, M. "Managing Organizational Knowledge in Framing the Domains of IT Management: Projecting the Future through the past," in: *Pinnaflex Educational Resources, Cincinnati, OH*, 1999, pp. 15-28.

- Anonymous. "The Benefits of Visual Rules in Application Development," White paper. Innovations Software Technology Bosch Group.
- Baldwin, C.Y., and Clark, K.B. "Managing in an Age of Modularity," *Harvard Business Review* (75:5), 1997, pp. 84-93.
- Baldwin, C.Y., and Clark, K.B. *Design Rules, Volume I, the Power of Modularity*. MIT Press, Cambridge, MA, 2000.
- Barrett, R. "People and Policies: Transforming the Human-Computer Partnership," in *Proceedings of the Fifth IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY'04)*, 2004.
- Bednar, C. "Capturing and Packaging Knowledge," in: *The Knowledge Management Yearbook 1999-2000*, J.W. Cortada and J.A. Woods (eds.), 1999, p. 211.
- Cooper, B.L., Watson, H.J., Wixom, B.H., and Goodhue, D.L. "Data Warehousing Supports Corporate Strategy at First American Corporation," *MIS Quarterly* (24:4), December 2000, pp. 547-567.
- Davenport, T., De Long, D., and Beers, M. "Building Successful Knowledge Management Projects," *Sloan Management Review* (39:2), 1998, pp. 43-57.
- Duchessi, P., and O'Keefe, R.M. "Understanding Expert Systems Success and Failure," *Expert Systems with Applications* (9:2), 1995, pp. 123-133.
- Evelson, B., Teubner, C., and Rymer, J.R. "How the Convergence of Business Rules, BPM, and BI Will Drive Business Optimization," *Forrester Research*, May 14, 2008.
- Galunic, C., and Weeks, J. "Managing Knowledge at Booz-Allen & Hamilton: Knowledge On-Line and Off," INSEAD Cases, Fontainebleau, France, 1999.
- Gartner, Report from Gartner Business Process Management Summit, Las Vegas, NV, 2008.
- Gill, T.G. "Early Expert Systems: Where are They Now?" *MIS Quarterly* (19:1), March 1995, pp. 51-81.
- Gruber, T.R. "Toward Principles for the Design of Ontologies Used for Knowledge Sharing," *International Journal of Human Computer Studies* (43:5), 1995, pp. 907-928.
- Hackbarth, G., and Grover, V. "The Knowledge Repository: Organizational Memory Information Systems," *Information Systems Management* (16:3), 1999, pp. 21-30.
- Hendriks, P. "Why Share Knowledge? The Influence of ICT on the Motivation for Knowledge Sharing," *Knowledge and Process Management* (6:2), 1999, pp 91-100.
- Hoffer, J.A., Prescott, M.B., and Topi, H. *Modern Database Management*, (9th ed.) Prentice-Hall, Upper Saddle River, NJ, 2009.
- Inmon, W.H. *Building the Data Warehouse*, John Wiley & Sons, 2005.
- Jeng, J.J., Flaxer, D., Kapoor, S. "RuleBAM: A Rule-Based Framework for Business Activity Management," in *Proceedings of 2004 IEEE International Conference on Services Computing (SCC'04)*, 2004, pp.262-270.
- Mao, J.-Y., and Zhang, X. "End-To-End User Participation in Information Systems Development: A Case Study Based on a Control Perspective," in *Proceedings of the International Conference on Information Systems (ICIS 2008)*, Paris, France, 2008.
- Markus, M. "Toward a Theory of Knowledge Reuse: Types of Knowledge Reuse Situations and Factors in Reuse Success," *Journal of Management Information Systems* (18:1), 2001, pp. 57-93.
- Morgan, T. *Business Rules and Information Systems: Aligning IT with Business Goals* Addison-Wesley Professional, 2002.
- Myers, P., and Swanborg Jr, R. "Packaging Knowledge," *The Knowledge Management Yearbook (1999-2000)* 1999, p. 200.
- Nagl, C., Rosenberg, F., Dustdar, S. "VIDRE--A Distributed Service-Oriented Business Rule Engine based on RuleML," in *Proceedings of the 10th IEEE International Enterprise Distributed Object Computing Conference (EDOC'06)*, 2006, pp.35-44.
- Nonaka, I. "A Dynamic Theory of Organizational Knowledge Creation," *Organization science* (5:1), 1994, pp. 14-37.
- Ross, R.G. "The Business Rule Book: Classifying, Defining, and Modeling Rules," Boston: Database Research Group Inc, 1994.
- Ross, R.G. *Principles of the Business Rule Approach*, Addison-Wesley Professional, 2003.
- Roth, G., and Kleiner, A. "Developing Organizational Memory Through Learning Histories," *Organizational Dynamics* (27:2) 1998, pp 43-60.
- Rymer, J.R., and Gualtieri, M. "The Forrester Wave™: Business Rules Platforms, Q2 2008," Forrester Research. 2008.
- Simon, H.A. "The Architecture of Complexity," *Proceedings of the American Philosophical Society*, 1962, pp. 467-482.

- Steinke, G., and Nickolette, C. "Business Rules As the Basis of an Organization's Information Systems," *Industrial Management and Data Systems* (103:1), 2003, pp. 52-65.
- Turban, E., Aronson, J., Liang, T., and Sharda, R. *Decision Support and Business Intelligence Systems*, Prentice-Hall, Inc. Upper Saddle River, NJ, USA, 2006.
- Valacich, J.S., George, J., and Hoffer, J.A. *Essentials of Systems Analysis and Design*, Prentice Hall Press, 2009.
- Von Halle, B. *Business Rules Applied*. John Wiley & Sons, New York, 2002.
- Von Halle, B., Goldberg, L., and Zachman, J. *Business Rule Revolution: Running Business the Right Way*. Happy About, 2006.
- Wan-Kadir, W.M.N., and Loucopoulos, P. "Relating Evolving Business Rules to Software Design," *Journal of Systems Architecture* (50:7), 2004, pp. 367-382.